Efficient *k*-NN Searching over Large Uncertain Time Series Database

Ailing Qian
School of Mathematics and Information Engineering
Taizhou University, 605 Dongfang Avenue Linhai, Taizhou, China
and School of Computer Science and Technology
Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan, China
Email: alinghb@126.com

Xiaofeng Ding and Yansheng Lu
School of Computer Science and Technology
Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan, China
Email: {xfding, lys}@hust.edu.cn

The k *nearest neighbor search over time-series databases has been a hot research topic for a long time period, which is widely used in information retrieval, genetic data matching, data mining, and so on. However, due to high dimensionality (i.e. length) and uncertainty of the time series, the* k *nearest neighbor search over directly indexed precise time series usually encounters serious problems, such as the "dimensionality curse" and "trust ability curse". Conventionally, many dimensionality reduction techniques and uncertainty processing strategies are proposed separately to break such drawbacks by reducing the dimensionality of time series and simulating the data uncertainty. However, among all the proposed methods, there are no indexing mechanisms to support* k *nearest neighbor queries over very large uncertain time-series databases. In this paper, we re-investigate PLA for approximating and indexing uncertain time series. In particular, we propose a novel distance function in the reduced PLA-space, and this function leads to a lower bound of the Euclidean distance between the original uncertain time series, which can lead to no false negatives during the search. In the following, based on five lemmas, we develop an effective approach to index these lower bounds using R-tree to improve the* k *nearest neighbor query efficiency. Finally, extensive experiments over synthetic data sets have been conducted to demonstrate the efficiency and effectiveness of PLA together with the proposed lemmas, in terms of both pruning power and wall clock time, compared with the baseline algorithm.*

Keywords: *time series; uncertain database; k nearest neighbor search; similarity search*

ACM Classification: *H.2.8*

1. Introduction

Since the early 1990s, the retrieval of *k* nearest neighbor time series has been widely studied, and this topic remains a hot research direction nowadays due to its wide usage in many applications (Chen *et al*, 2007; Kanth *et al*, 1998; Korn *et al*, 1997; Beskales *et al*, 1997; Kwang, 2006), such as

Manuscript received: 1 February 2012
Communicating Editor: Jiuyong Li

sensor network monitoring (Cranor *et al*, 2003), moving object tracking (Chen *et al*, 2005), and stock data analysis. In a coal mine application [1, 6], for example, sensors are deployed in the underground to collect data such as temperature and density of oxygen, which can be modeled as time series. Since emergency events usually correspond to some specific patterns, the event detection can be considered as the pattern search over time series data, which demands fast retrieval and quick reaction to keep the safety of coal miners. In this paper, we reinvestigate the *k* nearest neighbor search problem that obtains nearest candidates which are similar to a given query in a time-series database, especially in the case where the total number of time series in the database is large and each time series is uncertain (i.e. with different values for one record).

Uncertainty is ubiquitous in many emerging applications, and there are many information systems that need to cope with time series containing uncertain data. Consider in a weather monitoring system, many sensors are deployed to gather the temperature and moisture of some places, due to the errors caused by themselves, one unique time series that specifies the possible mistakes is provided by each sensor, that is to say, the returned values for even one place is not consistent to each other but are given by a group of probable observations.

Different from the traditional similarity search[1] problem over precise time-series database, data uncertainty makes the query result directly retrieved from those precise values meaningless (Cheng et al, 2004; Lian and Chen, 2008). That is, we cannot simply label a time series with YES or NO to be in the query result, but return those time series in the query result together with their nearest neighbor probabilities.

Since the length of uncertain time series is usually very long (e.g. larger than 1024), it becomes infeasible to index time series directly using spatial indexes, such as U-tree (Tao *et al*, 2005). This is caused by the serious “dimensionality curse” problem in high dimensional space. Generally speaking, when the dimensionality becomes very high, the query performance of the similarity search based on a multidimensional index can be even worse than that of a baseline approach (e.g. linear scan).

Motivated by this, our work introduces a new index structure, based on the well know PLA index, for answering *k* nearest neighbor queries over uncertain time-series efficiently under the transformed data space. The main contributions of this paper are summarized as below:

- For the first time, we give the formal definition of probabilistic *k* nearest neighbor queries over uncertain time series.
- Based on the transformed uncertain time series in *PLA* space, we propose five lemmas that can be used to speed up the query efficiency.
- We propose the *p*-*k*-NN query processing algorithm which can integrate the R-tree and prune lemma seamlessly.
- Extensive experiments have been made to verify the efficiency and effectiveness of our proposed query processing algorithm.

The rest of the paper is organized as follows. Section 2 reviews previous work on similarity search. Section 3 introduces the problem definition and shows the basic solution. Section 4 gives our new lower bound distance function for PLA, and illustrates the proposed query processing algorithm with five lemmas. Section 5 demonstrates the experimental results. The paper concludes with Section 6.

[1] Note that, the words similarity search and nearest neighbor query are used interchangeably, since they are assumed to have the same meaning for simplicity.

2. Related Work

In the last few years, many studies on similarity search over time-series have been conducted thoroughly in the database community.

The majority of current works are focused on two aspects: new dimensionality reduction techniques and new approaches to measure the similarity between two time series. In particular, the representatives of existing dimensionality reduction techniques include DFT (Agrawal, 1993), Piecewise Linear Approximation (PLA) (Morinaka, 2001), Chebyshe Polynomials (CP) (Cai and Ng, 2004), and Discrete Wavelet Transform (DWT) (Popivanov and Miller, 2002; Wu *et al*, 2000). CP is the most costly method in terms of time complexity, and PLA is much lower. The widely used similarity measurement functions include L_p-norm (Yi and Faloutsos, 2000), Edit distance with Real Penalty (ERP) (Chen and Ng, 2004), and Edit distance with Real Sequence (EDR) (Chen *et al*, 2005).

For nearest neighbors search, the reduction methods are first used to reduce the dimensionality of each time series to a lower dimensional space, and then apply a new metric function to obtain the similarity between two transformed data. Generally speaking, to guarantee there are no false dismissals, the distance between compressed sequences in the reduced space should be smaller than that of their Euclidean distance in original space.

To the best of our knowledge, previous work on the similarity search without false dismissals only deals with precise time series, which however cannot be used directly to uncertain time series. Thus, in this paper, we propose new techniques that can deal with similarity search over uncertain time series. Extensive experiments are also conducted to verify the effectiveness and efficiency of our proposed methods.

3. Problem Definition

In this section, we first give the definition of probabilistic *k* nearest neighbor queries in an uncertain time series database *D*, and then describe the baseline approach.

A. Probabilistic *k*-NN search

For a tuple *TS* recorded in an uncertain time-series database, it has the form of $<s_{1l}, s_{1u}; s_{2l}, s_{2u}; \ldots ; s_{nl}, s_{nu}>$, where s_{il}, s_{iu} ($1 \leq i \leq n$) are the lower bound and upper bound of s_i respectively, and *n* is the length of time-series TS_u. The distribution of s_i within the closed region (s_{il}, s_{iu}) follows a non-zero probability density function (*pdf*) such as uniform. Assume we have an uncertain time-series database *D*, in which each uncertain time series resides within its uncertainty region following some *pdf* (i.e. Uniform). We assume that all the time series are independent of each other in *D* similar to conventional works. Then, the problem of retrieving *k*-NN can be defined as follows:

Definition 1: Given an uncertain time series database *D* and a query time series *Q*, a probabilistic *k* nearest neighbors query (*p*-*k*-NN) retrieves *k* uncertain time series TS_1, TS_2, …, TS_k such that: $Pr_{nn}(TS_1, Q) \geq Pr_{nn}(TS_2, Q) \geq \ldots Pr_{nn}(TS_k, Q)$, and there are no time series from $D \setminus \{TS_1, TS_2, \ldots, TS_k\}$ having their NN probabilities greater than $Pr_{nn}(TS_k, Q)$. The probability of *TS* as the nearest neighbor $Pr_{nn}(TS, Q)$ is defined as follows:

$$Pr_{nn}\ (TS, Q) = \int_n^f \{Prob(Dist(TS,Q) = r) \times \prod_{\forall TS_a \in D \setminus \{TS\}} Prob(Dist(TS_a, Q) \geq r)\} dr \qquad (1)$$

That is to say, the probabilistic *k*-NN query is a probabilistic nearest neighbor query, which returns those uncertain time-series that have the smallest distance to a given query time-series *Q* with probabilities ranking at top-*k*. In particular, *n* and *f* are the minimum and maximum distances of *TS* from *Q* respectively.

The query time-series can either be precise or uncertain data, that is, $Q = <q_1, q_2, \dots, q_n>$ or $Q = <q_{1l}, q_{1u}; q_{2l}, q_{2u}; \dots; q_{nl}, q_{nu}>$. In order to measure the similarity between two uncertain time series, a distance function *Dist*(*TS*, *Q*) is needed for measuring such uncertain time series. As we know, any L_p-norm function can be used to measure the distance between pairs of precise time series. However, due to the uncertainty of the time series, the distance between them is also uncertain. Instead of obtaining one unique distance value between the corresponding time series, the distance of two uncertain time series rather consists of several distance values reflecting the distribution of all possible distance values between the instances of the corresponding uncertain time series. Thus, the distance between *TS* and *Q* is formalized in the following definition:

$$Dist^2(TS, Q) = \sum_{i=1}^{n} (s_i - q_i)^2, s_i \in [s_{il}, s_{iu}]. \quad (2)$$

In this paper, we are targeted to give the solution of *k*-NN to a precise query time-series *Q* and our method will be extended to solve the uncertain one in the future. The commonly used symbols in this work are summarized in Table 1.

Notation	Meaning
$\|TS\|$	the length of uncertain time series *TS*
s_i	the value at timestamp *i* in time series *TS*
$Dist(TS_1, TS_2)$	the Euclidean distance between time-series TS_1 and TS_2
$Dist_{PLA}(TS_1, TS_2)$	the distance between two time-series TS_1 and TS_2 in the reduced *PLA* space
D	time series database
Q	query time series
Prob(*)	the probability of some distances or events happening
$Pr_{nn}(TS, Q)$	the probability of *TS* as the NN of *Q*

Table 1: Frequently Used Symbols

B. Baseline Approach

Since there is no previous work on *k*-NN query over uncertain time series, the only available solution is adopting linear scan, which directly applies the basic distance calculating method Equation (2) on every time series, and then using the cost expensive Equation (1) to get their probabilities as the nearest neighbor. However, it is inefficient to scan all the records in *D* and calculate their nearest neighbor probabilities involving integral computation, as will be shown in the experiments section.

4. Methodology

In this section, we first introduce the dimensionality reduction techniques that do not introduce any false negatives while filtering through the index. Second, we give the pruning lemmas and illustrate the *k* nearest neighbor query processing algorithm, which efficiently returns *k* time series from a time-series database that have the smallest distances to a given query time series with top-*k* probabilities. Note that we are focusing on whole matching using Euclidean distance in this paper, the techniques can be extended to handle subsequence matching.

A. Rationale about PLA

Since the length of uncertain time-series is too large, it is impossible to index these time-series using a spatial index like R-tree (Guttman, 1984) or U-tree (Tao *et al*, 2005) directly, so dimensionality reduction techniques should be used to reduce the original uncertain time-series into a lower dimensional space. For these proposed dimensionality reduction techniques we are going to use *PLA* for its high reconstruction accuracy and strong pruning power.

For uncertain time series TS_u = <s_{1l}, s_{1u}; s_{2l}, s_{2u}; … ; s_{nl}, s_{nu}>, it is easy to be divided into two time-series with only lower bound and upper bound respectively: TS_{lu} = <s_{1l}; s_{2l};…; s_{nl}> and TS_{uu}= <s_{1u}; s_{2u};…;s_{nu}>. For each time series TS_{lu}, TS_{uu} and Q with length n, we divide each of them into m (m = n/l) non-overlapping segments of equal length l, then the *PLA* technique can be easily conducted according to the law given in Chen *et al* (2007). Therefore, the *PLA* representation of the above mentioned time series can be constructed as below:

$$TS_{lu\text{-}PLA} = \langle a_{11}, b_{11}; a_{12}, b_{12}; \dots ; a_{1m}, b_{1m}\rangle$$
$$TS_{uu\text{-}PLA} = \langle a_{21}, b_{21}; a_{22}, b_{22}; \dots ; a_{2m}, b_{2m}\rangle$$
$$Q_{PLA} = \langle a_{31}, b_{31}; a_{32}, b_{32}; \dots ; a_{3m}, b_{3m}\rangle$$

Thus, the lower bound distance between $TS_{lu\text{-}PLA}$ and Q_{PLA} is defined by:

$$\begin{aligned} Dist^2_{PLA}(TS_{lu}, Q) &= \sum_{i=1}^{m}\sum_{j=1}^{l}[(a_{1i}-a_{3i})j+(b_{1i}-b_{3i})]^2 \\ &= \sum_{i=1}^{m}\Big(\frac{l(l+1)(2l+1)}{6}(a_{1i}-a_{3i})^2+l(l+1)(a_{1i}-a_{3i})\times \\ &\qquad (b_{1i}-b_{3i})+l(b_{1i}-b_{3i})^2\Big) \end{aligned}$$

Similarly, the lower bound distance between $TS_{uu\text{-}PLA}$ and Q_{PLA} is:

$$\begin{aligned} Dist^2_{PLA}(TS_{uu}, Q) &= \sum_{i=1}^{m}\sum_{j=1}^{l}[(a_{2i}-a_{3i})j+(b_{2i}-b_{3i})]^2 \\ &= \sum_{i=1}^{m}\Big(\frac{l(l+1)(2l+1)}{6}(a_{2i}-a_{3i})^2+l(l+1)(a_{2i}-a_{3i})\times \\ &\qquad (b_{2i}-b_{3i})+l(b_{2i}-b_{3i})^2\Big) \end{aligned}$$

On the other hand, the squared Euclidean distance between TS_{lu} and Q is defined by:

$$\begin{aligned} Dist^2(TS_{lu}, Q) &= \sum_{i=1}^{n}[(a_{il}-q_i)]^2 \\ &= \sum_{i=1}^{m}\sum_{j=(i-1)+1}^{i-l}[(a_{jl}-q_j)]^2 \end{aligned}$$

Similarly, the squared Euclidean distance between TS_{uu} and Q is:

$$\begin{aligned} Dist^2(TS_{uu}, Q) &= \sum_{i=1}^{n}[(a_{iu}-q_i)]^2 \\ &= \sum_{i=1}^{m}\sum_{j=(i-1)+1}^{i-l}[(a_{ju}-q_j)]^2 \end{aligned}$$

Then the lower bound lemma holds for both sequence TS_{lu} and TS_{uu}:

$$Dist^2_{PLA}(TS_{lu}, Q) \leq Dist^2(TS_{lu}, Q)$$

$$Dist^2_{PLA}(TS_{uu}, Q) \leq Dist^2(TS_{uu}, Q)$$

Accordingly, we get a lower bound and an upper bound of the *PLA* distance between TS_u and Q respectively, which is:

$$Dist^2_{PLA}(TS_{lu}, Q) \leq Dist^2_{PLA}(TS_u, Q) \leq Dist^2_{PLA}(TS_{uu}, Q)$$

Therefore, the probability of *TS* as the nearest neighbor $Pr_{nn}(TS, Q)$ can be re-defined in the *PLA* space as follows:

$$Pr_{nn}\ (TS, Q) = \int_n^f \{Prob(Dist_{PLA}(TS,Q) = r) \times \prod_{\forall TS_a \in D \setminus \{TS\}} Prob(Dist_{PLA}(TS_a,Q) \geq r)\}dr \qquad (3)$$

Note that, *n* and *f* are the minimum and maximum distances of *TS* from *Q* in *PLA* space respectively. The property of *PLA* can guarantee no false negatives when pruning those unqualified time series through the index structure constructed in the reduced *PLA*-space. So the following important phases are how to prune those unqualified time series as much as possible and which kind of index method should be adopted to support efficient *k*-NN query processing.

B. Pruning Lemmas

Formally, given an uncertain time-series TS_u and a query Q in the *PLA*-space, the minimum (maximum) squared distance between them is either $dist^2_{PLA}(\mathrm{TS}_{lu}, Q)$ or $dist^2_{PLA}(TS_{uu}, Q)$. Note that the value comes from *m* disjoint segments, so for the *i*-th segment, the min (or max) value can be given by:

$$Dist^2_{PLA}(S_{i\text{-}lu}, Q_i) = \sum_{j=1}^{l}[(a_{1i} - a_{3i})j + (b_{1i} - b_{3i})]^2$$

$$Dist^2_{PLA}(S_{i\text{-}uu}, Q_i) = \sum_{j=1}^{l}[(a_{2i} - a_{3i})j + (b_{2i} - b_{3i})]^2$$

In the sequel, we will introduce several basic lemmas that make the foundation of the pruning approach, and then propose the technique which prunes those unqualified uncertain time series in the database that are definitely not *k*-NNs using the uncertain data model in Section 3.

Lemma 1: *Consider two uncertain time series S_u and T_u in the PLA-space, given a query time series Q, if the minimum squared distance value between S_u and Q, denoted as mindist$^2_{PLA}(S_u, Q_i)$, is larger than the maximum squared distance value maxdist$^2_{PLA}(T_u, Q_i)$ between T_u and Q , then S_u can be safely pruned.*

Proof: Obviously, because S_u can never be closer than T_u in the distance measurement, it has no chance of being the nearest neighbor of *Q*.

Note that those returned uncertain time series are considered as the candidate nearest neighbors of *Q* in the database. After the entire NN candidate set is selected from the reduced *PLA* space. It is easy to calculate the exact NN probabilities of each candidate within the set. We do not show the details here. Then, according to the following lemmas we can get the lower and upper distance bounds of the qualified nearest neighbors in *PLA* space.

Lemma 2: *The lower bound distance D_l of the nearest neighbors from query time series Q equals to the smallest minimum distances D_{smin} of all uncertain time series in D from Q.*

Proof: By contradiction. Suppose the lower bound distance of the *i*-th nearest neighbor is smaller than D_l, that is, $D_{il} < D_l$. However, note that, according to the definition of D_l, there are no time series within the database having their distance smaller than D_l, thus it is impossible to get a time series as the *i*-th nearest neighbor to be closer than D_l from *Q*. So the lemma holds.

Lemma 2 points out the lower bound distance of the nearest neighbors from query time series *Q*, and the upper bound distance could be calculated by the following lemma:

Lemma 3: *The upper bound distance D_u of the nearest neighbors from Q equals to the smallest maximum distances D_{smax} of all uncertain time series from Q.*

Proof: By contradiction. Among those time series maximum distances from *Q*, we denote the *i*-th ranking distance as D_{imax}, then there are at least a number of *i* uncertain time series that are completely within the circle centered at *Q* with radius D_{imax}. Suppose the upper bound distance D_u is greater than D_{smax}, that is, $D_u = D_{imax} > D_{smax}$. Then, for any time series TS_a with maximum distance within $[D_{smax}, D_u]$, it must be at least (*i*+1)-th nearest neighbor of *Q*, this is contradictory to define TS_a as the nearest neighbor within D_u. So the lemma holds.

Based upon the above two lemmas we can get the lower and upper bound distances of the nearest neighbors, so we can easily obtain the following lemma:

Lemma 4: *For any uncertain time series TS_a with maximum distance D_{amax} from Q, if D_{amax} is within the interval $[D_l, D_u]$, then TS_a has a chance to be the nearest neighbor of Q.*

Proof: The proof is obvious according to lemma 2 and lemma 3, and we do not show the details here.

Note that, in many applications, the query users are not concerned about the exact nearest neighbor probabilities values, they usually need answers that have larger confidence than a predefined threshold. In this work, we are focusing on nearest neighbors with top-*k* probabilities, the predefined probability threshold will be considered in our future work.

Thus, based on Lemma 1 and Lemma 4, our pruning approach which eliminates all the uncertain time series that have zero probability to be the nearest neighbor can be generalized as follows:

Lemma 5: *(distance pruning). Given a circle C (Q, D_u) which is centered at query Q with radius D_u and an uncertain time series TS_a with uncertain distance D_a from Q. The distance pruning method can safely prune TS_a if the minimum value of D_a is out of the circle C(Q, D_u), that is to say, is larger than D_u.*

Proof: According to definition of D_u, there exist at least one uncertain time series with its entire uncertainty region completely contained in circle *C* (*Q*, D_u). If the minimum value of TS_a does not intersect with circle *C* (*Q*, D_u), that is to say, the minimum distance n_a of TS_a from *Q* is larger than

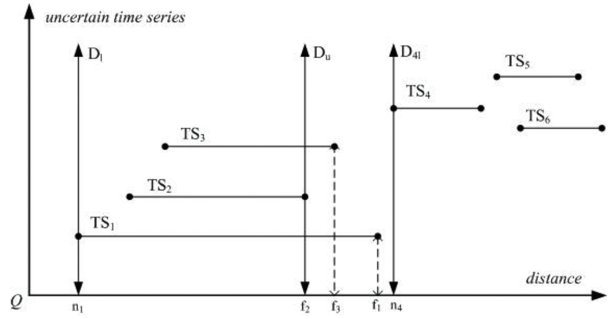


Figure 1: The Illustration of Distance Pruning

D_u. There always exist at least one time series with their distances to Q smaller than that of TS_a to Q, thus TS_a is at least the second nearest neighbor of Q with a certain probability. Therefore TS_a has no chance to be the nearest neighbor of Q, and thus can be safely pruned.

To better understand this lemma, we will use the distance interval described in Figure 1 for illustration. In particular, the vertical axis represents uncertain time series and the horizontal one corresponds the distance of each uncertain time series from Q. From the figure, we find that there are three time series TS_1, TS_2, and TS_3 having their maximum distances f_1, f_2, and f_3 smaller than D_{4l} (n_4) the minimum distance of TS_4 from the query, and the minimum maximum distance D_u equals f_2. Moreover, TS_4 is the first NN to D_{4l}, thus TS_4 is the fourth NN candidate to Q as there are always 3 time series with their distances closer to Q. Note that, the calculation of $Pr_{1\text{-NN}}$ (TS_4, D_{4l}) is efficient compared to $Pr_{4\text{-NN}}$ (TS_4, Q) which involves combinations of three time series TS_1, TS_2, and TS_3.

According to the above lemmas, the key point for pruning is to find *minmaxdist*^{2}(), the minimum of the maximum distance value of the uncertain time-series in the database. For this purpose we are going to construct two 2-dimensional R-trees $\mathrm{R_l}$ and $\mathrm{R_u}$, which contain each segment's coefficients of the lower bound time-series and upper bound series respectively. For example, < $\mathrm{a_{11}}$, $\mathrm{b_{11}}$> can be considered as a 2-dimensional point and inserted into $\mathrm{R_l}$. < $\mathrm{a_{21}}$, $\mathrm{b_{21}}$> can be considered as a 2-dimensional point and inserted into $\mathrm{R_u}$.

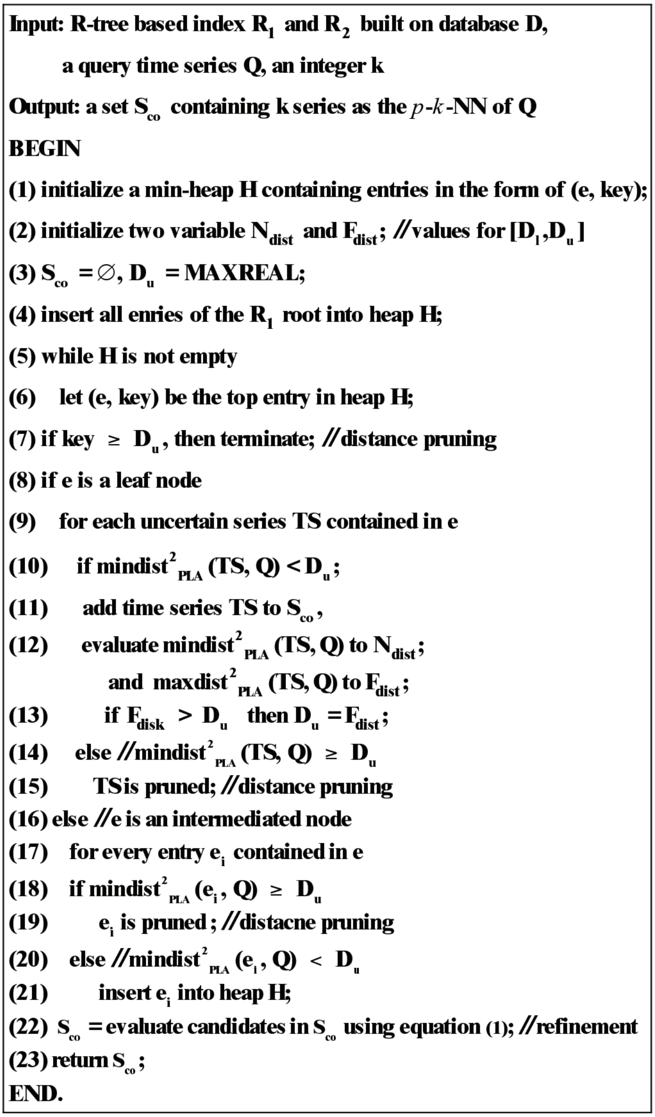
c. Index Structure and *p*-*k*-NN Algorithm

Recall that, the index divides each time series *TS* into *m* disjoint segments with equal length *l*, and for each segment, approximating it through two line segments with four coefficients <$\mathrm{a_i}$, $\mathrm{b_i}$> and <$\mathrm{a_j}$, $\mathrm{b_j}$>. Thus, each uncertain time series *TS* is transformed to totally 4*m* coefficients in the order of <$\mathrm{a_{1i}}$; $\mathrm{b_{1i}}$; $\mathrm{a_{1j}}$; $\mathrm{b_{1j}}$; ……>, which can be then divided into two 2*m*-dimensional points in the reduced space. In the sequel, we insert each transformed point into a 2*m*-dimensional index structure like R-tree (Guttman, 1984), on which the nearest neighbor search can be efficiently processed. Note that, since the index construction is the same as the standard R-tree, every entry of the nodes in the R-tree consists of an MBR (Minimum Bounding Rectangle) containing the reduced *PLA* data and a pointer indicates its corresponding sub-tree.

Obviously, different from the traditional nearest neighbor search in R-tree that uses Euclidean distance as the lower bound function, in the reduced *PLA* space, the index adopts the lower bound function as described above.

Figure 2 presents the detailed query procedure. Specifically, the procedure takes three parameters as input: two R-tree based index R_1 and R_2 constructed from an uncertain time series database *D* in the *PLA* space, a user specified query time series *Q*, and an integer *k* which indicates the number of query results. The procedure returns *k* uncertain time series as the *p*-*k*-NN query results.

In particular, we maintain a minimum heap *H* which contains entries in the form of (*e*, key), where *e* is the node MBR in the R-tree index, and key is the sorting key of the heap which is defined as the minimum distance of the MBR node to query time series *Q*. In addition, two variables $\mathrm{N_{dist}}$ and $\mathrm{F_{dist}}$ are initialized with zero for storing the values of bounding region [D_l, D_u]. Moreover, we also initialize a *p*-*k*-NN candidate set S_{co} to be empty and set variable D_u to MAXREAL in line 3, where D_u is the *k*-th smallest maximum distance to query series *Q* for all the time series that have been obtained so far. Then, we insert all the root entries of R-tree into heap *H* (line 4). While the heap *H* is not empty (line 5 – 21), we remove an entry (*e*, key) from the top of *H* (line 5), and check whether



```
Input: R-tree based index $R_1$ and $R_2$ built on database D,
        a query time series Q, an integer k
Output: a set $S_{co}$ containing k series as the $p$-$k$-NN of Q
BEGIN
(1) initialize a min-heap H containing entries in the form of (e, key);
(2) initialize two variable $N_{dist}$ and $F_{dist}$; //values for $[D_l, D_u]$
(3) $S_{co} = \varnothing$, $D_u$ = MAXREAL;
(4) insert all enries of the $R_1$ root into heap H;
(5) while H is not empty
(6)    let (e, key) be the top entry in heap H;
(7) if key $\geq$ $D_u$, then terminate; //distance pruning
(8) if e is a leaf node
(9)    for each uncertain series TS contained in e
(10)     if $mindist^2_{PLA}$ (TS, Q) < $D_u$;
(11)       add time series TS to $S_{co}$,
(12)       evaluate $mindist^2_{PLA}$ (TS, Q) to $N_{dist}$;
               and $maxdist^2_{PLA}$ (TS, Q) to $F_{dist}$;
(13)          if $F_{disk}$ > $D_u$ then $D_u = F_{dist}$;
(14)     else //$mindist^2_{PLA}$ (TS, Q) $\geq$ $D_u$
(15)        TS is pruned; //distance pruning
(16) else //e is an intermediated node
(17)    for every entry $e_i$ contained in e
(18)    if $mindist^2_{PLA}$ ($e_i$, Q) $\geq$ $D_u$
(19)         $e_i$ is pruned; //distacne pruning
(20)    else //$mindist^2_{PLA}$ ($e_i$, Q) < $D_u$
(21)         insert $e_i$ into heap H;
(22)  $S_{co}$ = evaluate candidates in $S_{co}$ using equation (1); //refinement
(23) return $S_{co}$;
END.
```

Figure 2: The Query Procedure for *p*-*k*-NN Processing

the key is greater than or equal to D_u. If it does, then it means that all the remaining entries in heap *H* would have their minimum distances to *Q* not smaller than D_u. Thus, according to the distance pruning approach they cannot contain any *p*-*k*-NNs and the procedure can be terminated.

5. Experimental Evaluations

In this section, through extensive experiments we illustrate the effectiveness of *PLA* together with our proposed bounding lemmas in the transformed *PLA*-space, in terms of both the pruning power and wall clock time. In particular, the pruning power is the fraction of time series that can be pruned in the reduced space, the wall clock time includes CPU time and I/O time (each I/O equals to 10ms). In order to test the efficiency of query processing, we use large synthetic data sets, each of which contains about 50K uncertain time series of length ranging from 128 to 1024. Based on the data sets with exact boundaries, we generated uncertain time series by generating samples uniformly distributed between the given values. We also used other distributions like Gaussian, but since our experimental results show that the sample distributions do not make much sense in the query processing procedure, we only show the results with uniform distribution.

The experiments were conducted on a computer with Intel Pentium(R) Dual-Core 2.50GHz & 2.52GHz CPU and 2048MB main memory running Window XP operating system. The programming language is using C++, and the algorithm is implemented in Microsoft Visual 6.0 environment. All the reported results are an average over 100 runs.

A. Performance over dimensionality

In this subsection we test the efficiency of our proposed method and linear scan over synthetic data set with different reduced dimensionalities. In particular, the reduced dimensionality is increased from 6 to 20. The default length of time series is set to 256, the default value of *k* is set to 10.

Figure 3(a) and 3(b) illustrate the pruning power and the wall clock time of *PLA* over synthetic data sets with various dimensionalities respectively. We can observe that, the pruning power (denoted as ratio of pruned time series) of linear scan method is zero, but most values of the *PLA* methods are larger than 0.5. For the dataset, the pruning power of *PLA* increases linearly when we increase the reduced dimensionality. This is reasonable, because high dimensionality makes the time series prone to be unqualified as some of its reduced dimensionalities do contribute lots of distances to the overall score. It is reported that high pruning power can be obtained with high reduced dimensionality. However the query performance over high dimensional indexes is not that efficient due to the "dimensionality curse". But the wall clock time of *PLA* as shown in Figure 3(b) is much smaller than that of the linear scan method, which indicates the efficiency of our proposed method.

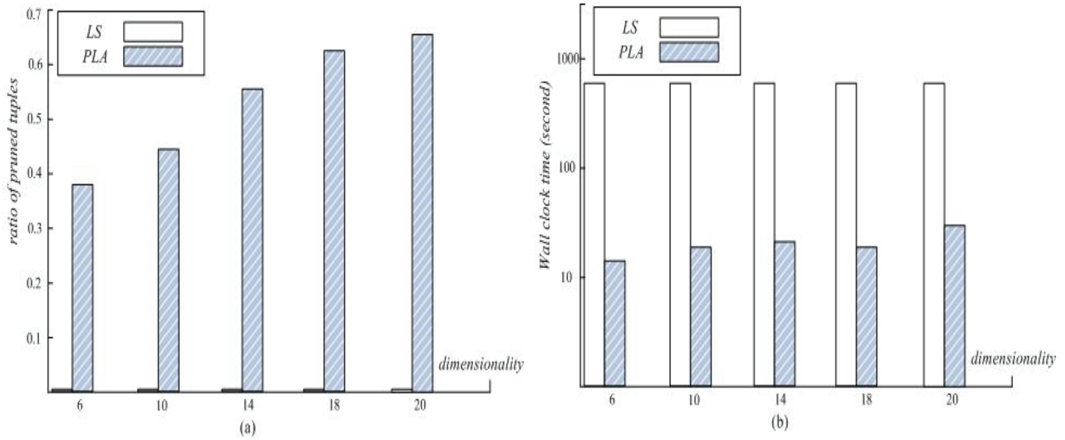


Figure 3: Performance vs. Reduced Dimensionality

B. Performance over Length

In this set of experiments, we test the efficiency of both methods over synthetic data set with different length of uncertain time series. Note that, the reduced dimensionality is set to 14 and the value of *k* is set to 10 by default.

As shown in Figure 4(a) and 4(b), the ratio of pruned time series in *PLA* is much larger than that of the linear scan, and the wall clock time of *PLA* is also much smaller than that of the base line approach. In particular, the pruning power in Figure 4(a) decreases when the length of time series increases. This is reasonable, as for the same dimensionality deduction method, the perturbation of longer time series will definitely greater than that of the shorter one, which would decrease the accuracy in the reduced spaces and thus lead to more pending tuples to be verified later. Furthermore, the wall clock time increase when the length of time series increases. This is as expected, since a larger length of time series requires more dimensionality reduction in the *PLA* space, which would need more computation to prune those unqualified time series.

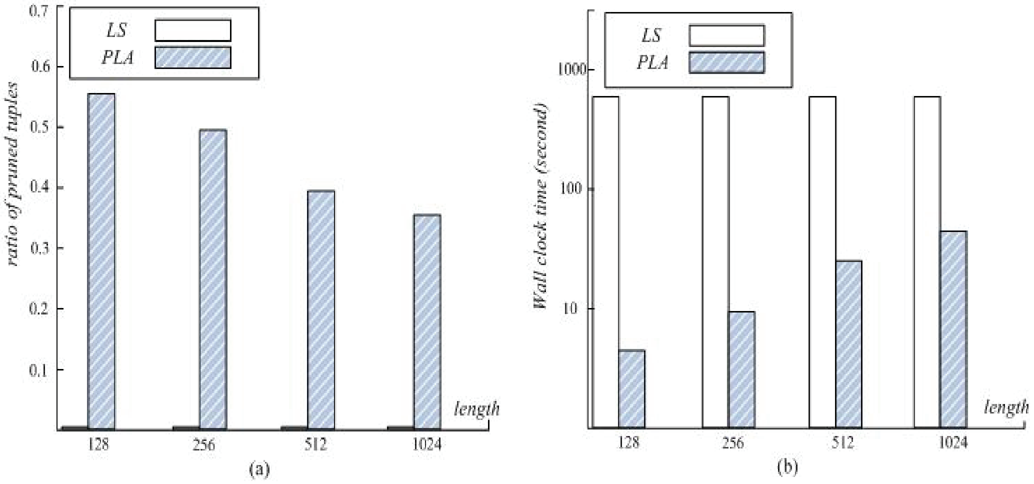
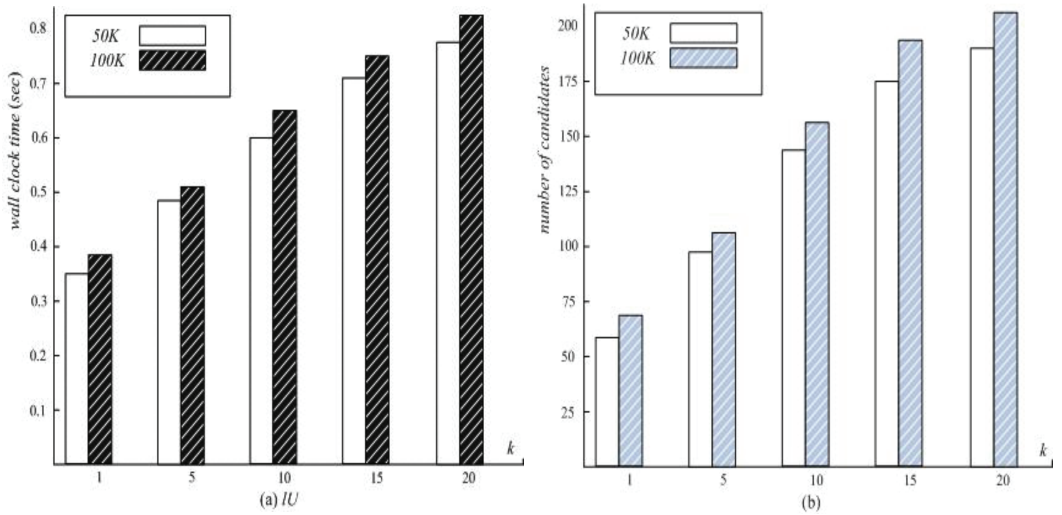


Figure 4: Performance vs. Length of Series

C. Performance over *k*

In this subsection, we evaluate the query performance of our proposed pruning approaches with respect to different *k* values specified by *p*-*k*-NN queries. In particular, the wall clock time and the number of candidates to be refined are compared over two synthetic data sets by varying *k* from 1 to 20. The data size equals to 50K and 100K respectively, the reduced dimensionality is set to 14 and the length of time series is set to 256 by default.

From the results as shown in Figure 5(a) and 5(b), we can see that, the wall clock time increases almost linearly with *k*. This is reasonable, since the number of candidates that after searching the R-tree based index structure using the distance pruning method grows with *k* (because the pruning distance D_u increases with *k*, thus more unqualified uncertain time series are unable to be eliminated), which is indicated by Figure 5(b), thus the wall clock time that used to search a number of increasing candidates must also go up. However, for a large dataset size 100K, as shown in both figures, the wall clock time and final number of candidates that have to be evaluated are



(a) *IU*

(b)

Figure 5: Performance VS. *k*

almost the same (a little higher) as that of the small size 50K at a specified parameter *k*, which indicates a good scalability of our proposed distance pruning approach.

6. Conclusions

Due to the inherent uncertainty of data in many emerging applications, query processing over these uncertain data becomes more and more important. The *k* nearest neighbor query is an important query type in traditional database, and it is widely used in Geographical Information Management Systems, Decision Supporting System and similarity search. However, previous *k* nearest neighbor query processing technique is not applicable in the context of uncertain time series database. This paper presents a novel *k* nearest neighbor query processing technique for uncertain time-series database. Using Euclidean distance in the *PLA* space as the similarity measurement and design effective distance pruning approaches to facilitate reducing the search space under the help of R-tree based index structure. Extensive experiments have been conducted to verify the efficiency and effectiveness of our proposed methods in terms of wall clock time, prune ratio and number of candidates to be refined, under synthetic data sets with various parameter settings.

In many applications the time series data is updated frequently, thus as an interesting direction, we will extend the solution to *p*-*k*-NN query processing techniques over databases with updates, that is to say, the query database *D* is no longer a static time series set, but with some its tuples are updatable. It would be challenging to obtain a trade off between the query efficiency and the index structure update performance.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 61100060, the China Postdoctoral Science Foundation funded project under grant No. 20100471179, and the Central Colleges of basic scientific research and operational costs under grant No. 2011QN054. Xiaofeng Ding is the corresponding author.

References

AGRAWAL, R., FALOUTSOS, C. and SWAMI, A.N. (1993): Efficient similarity search in sequence databases. In *FODO*.

BESKALES, G. , SOLIMAN, M.A. and ILYAS, I.F. (2008): Efficient Search for the Top-k Probable Nearest Neighbors in Uncertain Databases. In *Proc. Int'l Conf. Very Large Data Bases (VLDB)*, 326–339.

CAI, Y. and NG, R. (2004): Indexing spatio-temporal trajectories with Chebyshev polynomials. In *SIGMOD*.

CHEN, L. and NG, R. (2004): On the marriage of edit distance and Lp norms. In *VLDB*.

CHEN, L., OZSU, M.T. and ORIA, V. (2005): Robust and fast similarity search for moving object trajectories. In: *Proceedngs of ACM SIGMOD Int. Conf. on Management of Data*.

CHEN, Q., CHEN, L., LIAN, X., LIU, Y. and YU, J.X. (2007): Indexable PLA for efficient similarity search, In: *Proceedings of the 33st International Conference on Very Large Data Bases (VLDB)*. Vienna, Austria.

CHENG, R., KALASHNIKOV, D.V. and PRABHAKAR, S. (2004): Querying imprecise data in moving object environments. In *IEEE Transactions on Knowledge and Data Engineering*, 16(9): 1112–1127.

CRANOR, C.D., JOHNSON, T. and SPATSCHECK, O. (2003): Gigascope: A stream database for network applications. In: *Proceedngs of ACM SIGMOD Int. Conf. on Management of Data*.

GUTTMAN, A. (1984): R-trees: a dynamic index structure for spatial searching. In *SIGMOD*.

KANTH, K.V.R., AGRAWAL, D. and SINGH, A. (1998): Dimensionality reduction for similarity searching in dynamic databases. In: *Proceedings of ACM SIGMOD Int. Conf. on Management of Data*.

KORN, F., JAGADISH, H. and FALOUTSOS, C. (1997): Efficiently supporting ad hoc queries in large datasets of time sequences. In: *Proceedngs of ACM SIGMOD Int. Conf. on Management of Data*.

LIAN, X. and CHEN, L. (2008): Monochromatic and Bichromatic Reverse Skyline Search over Uncertain Database. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*. Vancouver, Canada. 213–226.

MORINAKA, Y., YOSHIKAWA, M., AMAGASA, T. and UEMURA, S. (2001): The L-index: An indexing structure for efficient subsequence matching in time sequence databases. In *PAKDD*.

PARK, K. (2006): An Efficient Data Dissemination Scheme for Nearest Neighbour Query Processing. *Journal of Research and Practice in Information Technology*, 38(2): 181–195, May.

POPIVANOV, I. and MILLER, R.J. (2002): Similarity search over time series data using wavelets. In *ICDE*.

TAO, Y., CHENG, R., XIAO, X., NGAI, W.K., KAO, B. and PRABHAKAR, S. (2005): Indexing multi-dimensional uncertain data with arbitrary probability density functions. In *VLDB*, 922–933.

WU, Y-L., AGRAWAL, D. and ABBADI, A.E. (2000): A comparison of DFT and DWT based similarity search in time-series databases. In *CIKM*.

XUE, W., LUO, Q., CHEN, L. and LIU, Y. (2006): Contour map matching for event detection in sensor networks. In: *Proceedings of ACM SIGMOD Int. Conf. on Management of Data*.

YI, B-K. and FALOUTSOS, C. (2000): Fast time sequence indexing for arbitrary Lp norms. In *VLDB*.

Biographical Notes

__Ailing Qian__ received her MS degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 2006 and her PhD degree in computer science from the same university in 2011. Currently, Dr Qian is an associate professor at the Taizhou University, China. Her major research interests include the time series data query processing and optimization, uncertain databases, data mining and wireless sensor network query systems.

Ailing Qian

__Xiaofeng Ding__ received his PhD degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2009. Currently, he is an associate professor in the School of Computer Science and Technology at Huazhong University of Science and Technology. His research interests include distributed computing, query processing, data privacy, uncertain data management, peer-to-peer computing and string databases.

Xiaofeng Ding

***Yansheng Lu** is currently a full professor and the head of the database and software engineering research group in the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. He has published more than 100 peer-reviewed papers in various journals and conference proceedings in the areas of database systems, web information systems, and software engineering. His main research interests include query processing in relational databases, graph mining, graph query processing, data stream processing and web services.*

Yansheng Lu